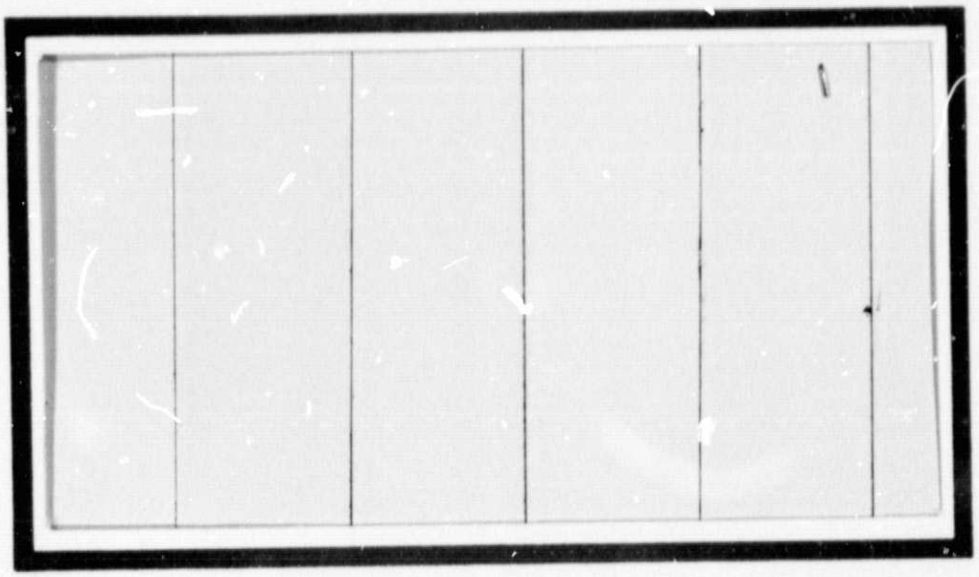
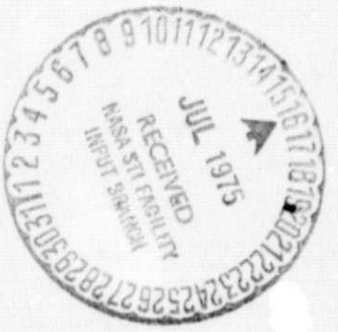
NASA CR-
141878



(NASA-CR-141878) LFSPMC: LINEAR FEATURE SELECTION PROGRAM USING THE PROBABILITY OF MISCLASSIFICATION (Texas A&M Univ.) 52 p HC $4.25 CSCL 12A

N75-26739

Unclas
G3/64 26682



DEPARTMENT OF MATHEMATICS

TEXAS A&M UNIVERSITY

COLLEGE STATION, TEXAS

LFSPMC: LINEAR FEATURE SELECTION PROGRAM USING THE PROBABILITY OF MISCLASSIFICATION

by

L. F. Guseman, Jr. and Bruce P. Marion

Department of Mathematics
Texas A&M University

Prepared For

Earth Observations Division
NASA/Johnson Space Center
Houston, Texas

Contract NAS-9-13894-1S

January, 1975

Report #3

TABLE OF CONTENTS

LIST OF FIGURES

1.0 INTRODUCTION

The problem of classification arises when an observer must determine the class of an object by investigating a set of measurements or features taken from the object. It is assumed that the object belongs to one of a finite number of classes (e.g. crops) and that each class is described by a probability distribution of its measurement vectors. When the dimension of the measurement vector is high and a large number of objects are to be classified the computational load increases significantly. As a result, one employs feature selection techniques which allow classification in spaces of lower dimension while preserving as much as possible the discriminatory power inherently available in the original measurements.

In the sequel we discuss the computational procedure and associated computer program for a linear feature selection technique. The technique assumes:

1. A finite number, m, of classes.
2. Each class is described by an n-dimensional multivariate normal density function of its measurement vectors.
3. The mean vector and covariance matrix for each density function are known (or can be estimated).
4. The a priori probability for each class is known.

The technique produces a single linear combination of the original measurements which minimizes the one-dimensional probability

of misclassification defined by the transformed densities.

The procedure for two classes with equal a priori probabilities was developed in [4]. Subsequent theoretical results from [5], which are summarized in Section 2.0, form the basis for the procedure described herein. The computational procedure and a description of the associated computer program appear in Section 3.0. Procedures for using the program and example input decks appear in Sections 4.0 and 5.0, respectively. Section 6.0 contains a sample output based on one of the examples from Section 5.0.

2.0 MATHEMATICAL PRELIMINARIES

Let $\Pi_1, \Pi_2, \ldots, \Pi_m$ be distinct classes (e.g. crops of interest) with known a priori probabilities $\alpha_1, \alpha_2, \ldots, \alpha_m$, respectively. Let $x = (x_1, x_2, \ldots, x_n)^T \in E^n$ denote a vector of measurements (e.g. ERTS multispectral scanner data from either a single pass or several registered passes) taken from an arbitrary element of $\bigcup_{i=1}^{m} \Pi_i$. Suppose that the measurement vectors for class Π_i are distributed according to the n-dimensional multivariate normal density function

$$p_i(x) = (2\pi)^{-n/2} |\Sigma_i|^{-1/2} \exp\left[-\frac{1}{2} (x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i) \right], \quad 1 \leq i \leq m .$$

We assume that the $n \times 1$ mean vector μ_i and the $n \times n$ covariance matrix Σ_i for each class Π_i are known with Σ_i positive definite, $1 \leq i \leq m$. The symbol $|A|$ is used to denote the determinant of the matrix A. The n-dimensional probability of misclassification, denoted by PMC, of objects from $\bigcup_{i=1}^{m} \Pi_i$ is given (see [1]) by

$$\text{PMC} = 1 - \int_{E^n} \max_{1 \leq i \leq m} \alpha_i p_i(x) dx$$

$$= 1 - \sum_{i=1}^{m} \alpha_i \int_{R_i} p_i(x) dx \quad ,$$

where the sets R_i, $1 \leq i \leq m$, called the Bayes' decision regions, are defined by

$$R_i = \left\{ x \in E^n : \alpha_i p_i(x) = \max_{1 \leq j \leq m} \alpha_j p_j(x) \right\} , \quad 1 \leq i \leq m .$$

The resulting classification procedure, called the Bayes' optimal classifier, is defined as follows:

Assign an element to Π_i if its vector x of measurements belongs to R_i, $1 \leq i \leq m$.

If $B = (b_1, \ldots, b_n)$ is a nonzero $1 \times n$ vector and $x \in E^n$, then $y = Bx \in E^1$ and the transformed measurements (i.e. $y = Bx$) for class Π_i are distributed according to the univariate normal density function (see [1]) given by

$$p_i(y,B) = (2\pi)^{-1/2} (B\Sigma_i B^T)^{-1/2} \exp\left[- \frac{(y - B\mu_i)^2}{2B\Sigma_i B^T} \right] , \quad 1 \leq i \leq m .$$

The probability of misclassification g of an object from $\bigcup_{i=1}^{m} \Pi_i$ in terms of the transformed measurement $y = Bx \in E^1$, as a function of nonzero B, is given by

$$g(B) = 1 - h(B)$$

$$= 1 - \int_{E^1} \max_{1 \leq i \leq m} \alpha_i \, p_i(y,B) \, dy$$

$$= 1 - \sum_{i=1}^{m} \alpha_i \int_{R_i(B)} p_i(y,B) \, dy ,$$

where the transformed Bayes' decision regions are given by

$$R_i(B) = \left\{ y \in E^1 : \alpha_i \, p_i(y,B) = \max_{1 \le j \le m} \alpha_j \, p_j(y,B) \right\} , \quad 1 \le i \le m \ .$$

We use h(B) to denote the probability of correct classification for B.

The computational procedure and associated computer program described in the sequel present a method for determining a nonzero $1 \times n$ vector B which minimizes g, or equivalently, which maximizes h. The method yields a linear feature selection procedure in that classification is ultimately performed in E^1 using only a single feature; namely, an optimal linear combination of the original measurements. The classification procedure in E^1 is described as follows:

> If B is a nonzero $1 \times n$ vector which minimizes g, then assign an object to Π_i if, for its measurement vector x, $Bx \in R_i(B)$.

It is readily verified that $h(tB) = h(B)$ for each scalar $t \neq 0$, and as a consequence the problem reduces to maximizing h over the compact set of $1 \times n$ vectors of norm one. The existence of a maximizing B follows from the continuity of h.

In seeking a maximum of h, it is natural to consider the differentiability of h with respect to the elements of B. In the sequel we make use of the Gateaux differential of h at B with increment C, denoted by $\delta h(B;C)$, and defined (if the limit exists) by

$$\delta h(B;C) = \lim_{s \to 0} \frac{h(B + sC) - h(B)}{s}$$

for a $1 \times n$ vector C. If, for a given nonzero B, the above limit exists for each $1 \times n$ vector C, then h is said to be Gateaux differentiable at B. Similarly, if B is a $1 \times n$ vector, we define (when the limit exists)

$$\delta p_i(y,B;C) = \lim_{s \to 0} \frac{p_i(y,B + sC) - p_i(y,B)}{s} ,$$

for a $1 \times n$ vector C. For an excellent discussion of Gateaux differentials see [7].

The computational procedure is based on the following theoretical results from [5].

Lemma. Let B be a nonzero $1 \times n$ vector. Then (omitting subscripts)

$$\delta p(y,B;C) = - p(y,B) \left[\frac{C\Sigma B^T}{B\Sigma B^T} - \frac{C\mu}{B\Sigma B^T} (y - B\mu) - \frac{C\Sigma B^T}{(B\Sigma B^T)^2} (y - B\mu)^2 \right]$$

for each $1 \times n$ vector C.

Theorem 1. Let B be a nonzero $1 \times n$ vector for which $\alpha_i p_i(y,B) \neq \alpha_j p_j(y,B)$ for $i \neq j$. Then h is Gateaux differentiable at B, and

$$\delta h(B;C) = \sum_{i=1}^{m} \alpha_i \int_{R_i(B)} \delta p_i(y,B;C)dy$$

Theorem 2. Let B be a nonzero $1 \times n$ vector at which h assumes a maximum. Then h is Gateaux differentiable at B.

By substituting the expression for $\delta p_i(y,B;C)$ given by the Lemma, into the expression from Theorem 1, and using integration by parts, we obtain the following result.

Theorem 3. Let B be a nonzero $1 \times n$ vector for which $\alpha_i p_i(y,B) \neq \alpha_j p_j(y,B)$ for $i \neq j$. Then h is Gateaux differentiable at B, and

$$\delta h(B;C) = - \sum_{i=1}^{m} \alpha_i p_i(y,B) \left[\frac{C\Sigma_i B^T}{B\Sigma_i B^T} (y - B\mu_i) + C\mu_i \right] \Bigg|_{R_i(B)} ,$$

where the notation $\Big|_{R_i(B)}$ denotes the sum of the values of the function at the right endpoints of the intervals comprising $R_i(B)$ minus the sum of its values at the left endpoint.

If B is a nonzero $1 \times n$ vector which minimizes $g(B) = 1 - h(B)$, then B must satisfy the vector equation

$$\frac{\partial g}{\partial B} \equiv \begin{pmatrix} \delta g(B;C_1) \\ \cdot \\ \cdot \\ \cdot \\ \delta g(B;C_n) \end{pmatrix} = \begin{pmatrix} 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{pmatrix} ,$$

where C_j, $1 \leq j \leq n$, is a $1 \times n$ vector with a one in the j^{th} slot

and zeros elsewhere. Using the formula for $\frac{\partial h}{\partial B}$ resulting from Theorem 3, and using the fact that $\frac{\partial g}{\partial B} = -\frac{\partial h}{\partial B}$, we obtain a numerically tractable expression for the variation in the probability of misclassification g with respect to B. The use of this expression in a computational procedure for obtaining a nonzero B which minimizes g is discussed in subsequent sections.

3.0 COMPUTATIONAL PROCEDURE

The computational procedure for determining the nonzero $1 \times n$ vector B which minimizes the probability of misclassification g with respect to the one-dimensional transformed density functions is embodied in the FORTRAN program LFSPMC. Included in the program is the capability to classify input measurement vectors using the computed B vector. In addition, the program provides the capability of estimating the probability of misclassification in the original feature space.

Apart from the various program parameters and option flags (discussed in the sequel), the basic input data to the program consists of the mean vectors and covariance matrices which comprise the class statistics deck. All input data to the program is from unit reference 5 (usually punched cards). All output from the program is printed on unit reference 6. Several additional options are built into the program which provide the user with the capability of making successive runs using designated subsets of the original classes or features already provided by the class statistics deck.

The program is divided into the following six subsections which are discussed in turn in the sequel:

Parameter Initialization

Initial Vector Determination

Optimization Algorithm

Computation of $g(B)$ and $\frac{\partial g}{\partial B}$

Classification Procedure

Estimation of PMC .

3.1 Parameter Initialization

All input variables to the program are of a fixed format and must be entered as shown in Section 4.0 and as illustrated in the examples in Section 5.0. These variables are:

M . . . . . : Number of classes, ≤MTOT.

N . . . . . : Dimension of feature space, ≤NFPC.

CLS . . . . . : Class names, 12 characters, double subscripted array.

MTOT. . . . . : Number of classes in the class statistics deck.

NFPC. . . . . : Number of features per class in the class statistics deck.

KCLS. . . . . : Numeric labels of the M designated classes from the MTOT classes in the class statistics deck, single subscripted array.

IFEA. . . . . : Numeric labels of the N designated features from the NFPC features in the class statistics deck, single subscripted array.

COVARB. . . . . : Input covariance matrices, triple subscripted array.

XMEANB. . . . . : Input mean vectors, double subscripted array.

APROB . . . . . : A priori probabilities for the M classes, single subscripted array.

A variety of flags must be initialized in the program during each run in order to select program options and establish program controls. These flags are designated as follows:

IST . . . . . : Class statistics deck input flag

= 0 input a new class statistics deck according to a specified format (discussed below).

= 1 use current class statistics deck.

IOP . . . . . : Computation flag

= 1 compute the minimizing B vector.

= 2 classify measurement vectors after computing the minimizing B vector.

= 3 classify measurement vectors using the current B vector

ICLSN . . . . . : PMC estimation flag

= 0 no computation of estimated PMC in N-dimensional space.

= 1 compute estimated PMC in N-dimensional space.

Parameter initialization (Fig. 1) is accomplished by entering M, N, CLS, IST, IOP, and ICLSN. If IST = 0, then MTOT and NFPC are entered for the new class statistics deck. If MTOT differs from M or NFPC differs from N, then the desired class and feature numbers, KCLS and IFEA, respectively, are entered. If equality between MTOT and M and between NFPC and N occurs, then the program sets the class and feature

numbers in ascending order. In either case, the corresponding mean vectors and covariance matrices are defined in XMEAN and COVAR respectively. The class statistics deck is comprised of the MTOT mean vectors in the order of ascending class numbers followed by the MTOT covariance matrices in the order of ascending class numbers. The entries of each mean vector in the order of ascending feature number are entered according to the format (5X, 5D15.8). The NFPC(NFPC + 1)/2 elements on and above the diagonal of each covariance matrix are entered by column in the format (5X, 5D15.8). It is assumed that the diagonal elements of each covariance matrix are in order of ascending feature number. The first entry of each new mean vector or covariance matrix starts on a new card. The values for APROB are entered for each run regardless of the value of IOP. The mean vector, covariance matrix, and a priori probability for each of the designated classes are printed. Control is then given to the minimization routine BVECT or to the classification routine CLSNEW according to the value of IOP. Computation of the estimated PMC in the original feature space is determined by the value of ICLSN.

SUBROUTINE PRDIM

SUBROUTINE BMAT

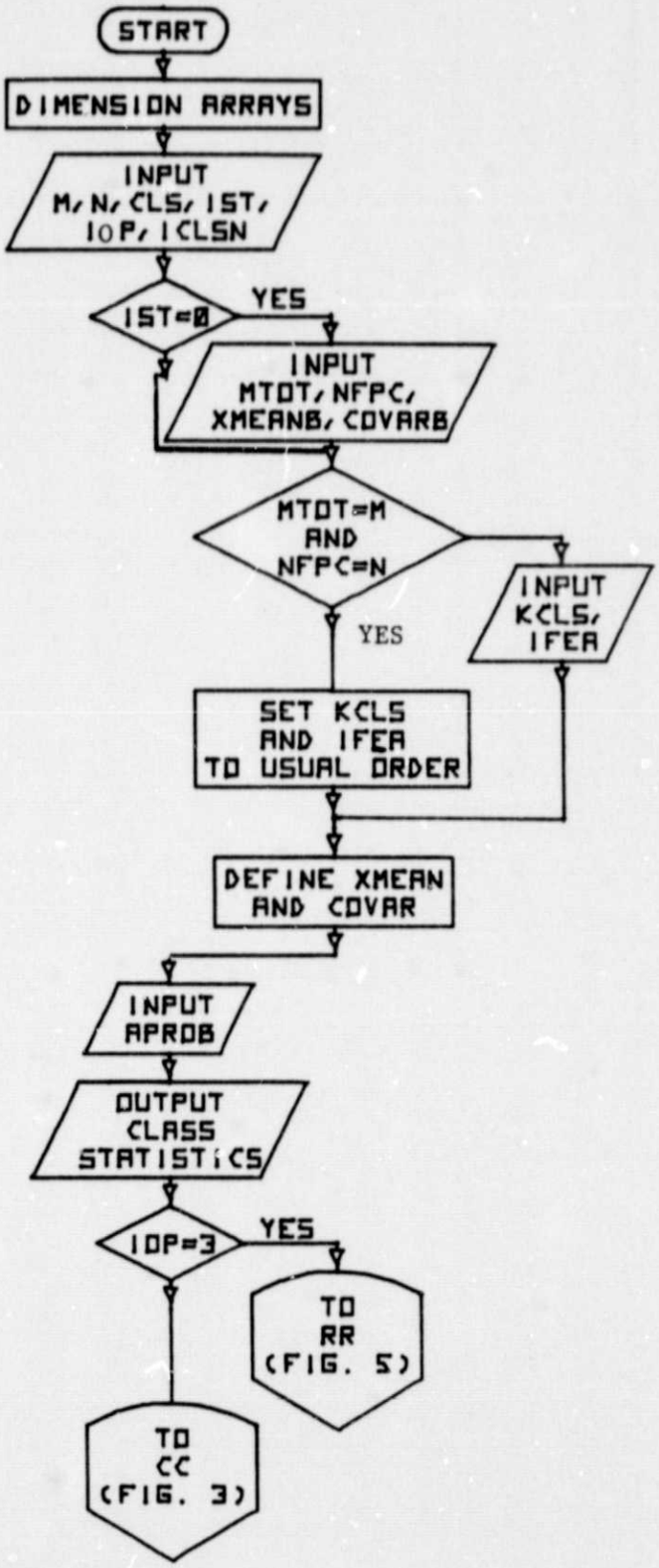


Fig. 1. Parameter Initialization Flowchart

3.2 Initial Vector Determination

A nonzero $1 \times n$ vector B which minimizes g cannot, in general, be obtained in closed form, and a numerical optimization (minimization) procedure is required. Any such optimization algorithm must be given an initial vector B_0. Two methods for determining initial vectors are provided by the input flag IZ.

IZ . . . . . : Initial vector flag

= 0 compute the initial vector B_0.

= 1 input the initial vector B_0.

When IZ = 0, the initial vector B_0 is computed within the program. For the special case of two classes with equal a priori probabilities, B_0 is computed in SUBROUTINE BC2CP using the formula (see [4], [5])

$$B_0 = (\mu_1 - \mu_2)^T (\Sigma_1 + \Sigma_2)^{-1} .$$

In all other cases the initial vector is computed in SUBROUTINE BCOMP using the procedure described below (see [9], [6]).

Given α_i, μ_i, and Σ_i, $1 \leq i \leq m$, let

$$\Sigma = \frac{1}{m} \sum_{i=1}^{m} \alpha_i \Sigma_i$$

and determine (using SUBROUTINE EIGEN*) an $n \times n$ matrix A such that $A \Sigma A^T = I$. Letting $\mu_i' = A\mu_i$, $1 \leq i \leq m$, the problem can then be

*Provided by TRW Systems, Inc. - Programmer: Jane Montgomery

reduced to finding a fixed point of the function G defined as follows:

For a given 1×n vector C, choose indices i_j, $1 \leq j \leq m$, for the μ'_i's and α_i's such that

$$C\mu'_{i_1} < C\mu'_{i_2} < \ldots < C\mu'_{i_m} .$$

For

$$a_j = \frac{\ln\left(\alpha_{i_j}/\alpha_{i_{j+1}}\right)}{C\left(\mu'_{i_{j+1}} - \mu'_{i_j}\right)} + \frac{C\left(\mu'_{i_{j+1}} + \mu'_{i_j}\right)}{2} , \quad 1 \leq j \leq m-1$$

we let

$$F(C) = \sum_{j=1}^{m-1} \alpha_{i_j} p_{i_j}(a_j, C) \left(\mu'_{i_{j+1}} - \mu'_{i_j}\right) .$$

Then

$$G(C) = \frac{F(C)^T}{||F(C)^T||_2} .$$

To find a fixed point of G (C=G(C)), we let $C_0 = \mu'_i - \mu'_j$, where

$$||\mu'_i - \mu'_j||_2 = \max_{r \neq s} ||\mu'_r - \mu'_s||_2 ,$$

and compute successive vectors C_k using the mean iteration formula (see [8])

$$C_{k+1} = \frac{k}{k+1} C_k + \frac{1}{k+1} G(C_k), \quad k = 0, 1, 2, \ldots .$$

The number of iterations is specified by the internal parameter ITER (25 is a reasonable value). Upon completion of the iterations, the final C_k, say C, is used to compute an initial vector B_0 from the formula

$$B_0 = CA \quad .$$

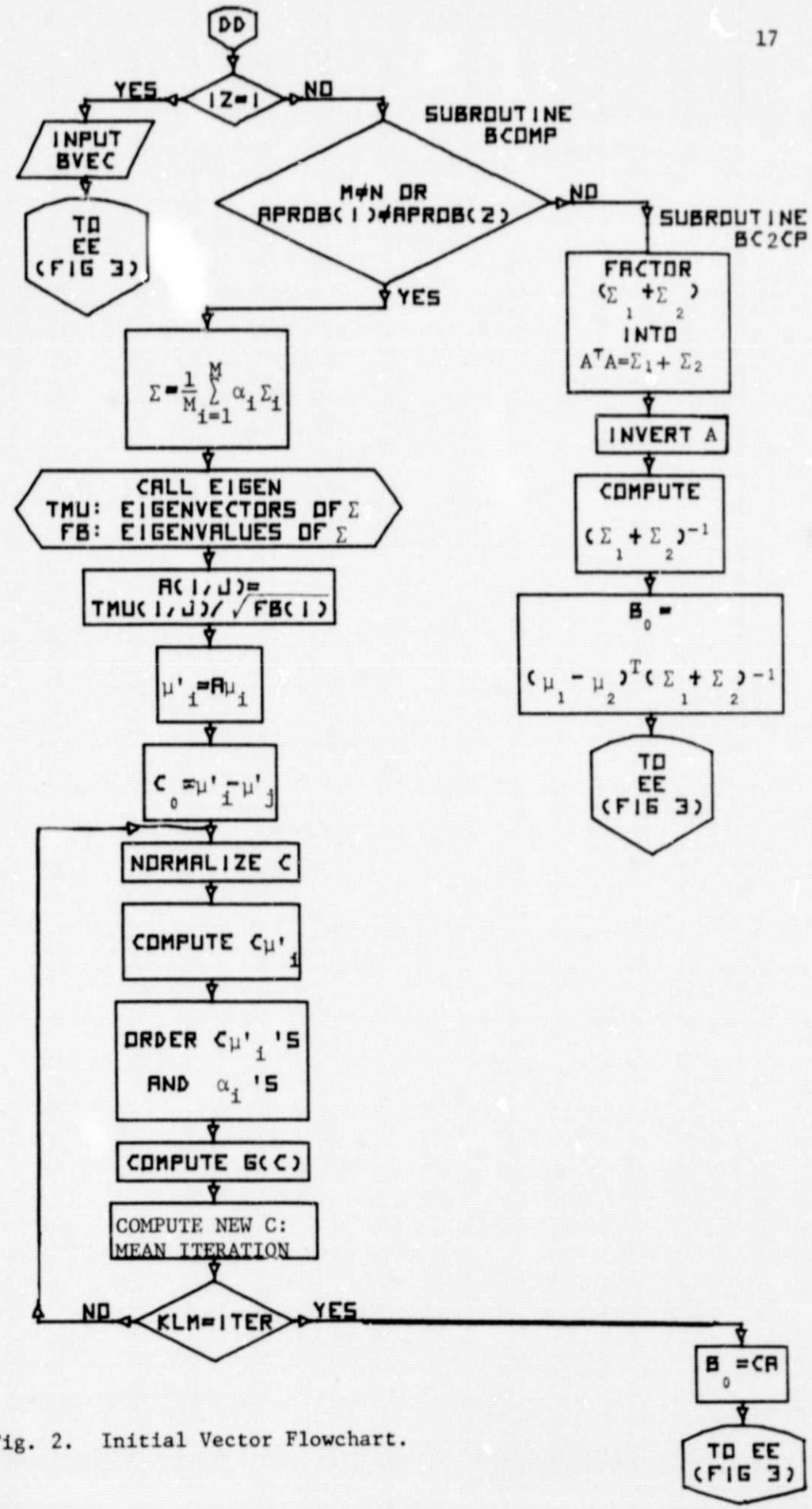


Fig. 2. Initial Vector Flowchart.

3.3 Optimization Algorithm

The numerical minimization algorithm used to find a local minimum of g is SUBROUTINE DFMFP from the IBM Scientific Subroutine Package [10]. The procedure is based on the method of Fletcher and Powell [3]. Computation of the minimizing B is controlled by SUBROUTINE BVECT which initializes the following input parameters used by DFMFP.

EST . . . . . : An estimate of the minimum value of g(B).

EPS . . . . . : Tolerance for the expected absolute error of the optimization algorithm. Experience has shown 10^{-4} to be a reasonable value.

LIMIT . . . . . : Maximum number of iterations for the optimization algorithm. Fifty iterations appears to be a reasonable value.

The iteration routine parameters may be defined internally if the user's application is of a repetitive nature. Calculation of g(B) and $\frac{\partial g}{\partial B}$ is discussed in the following subsection.

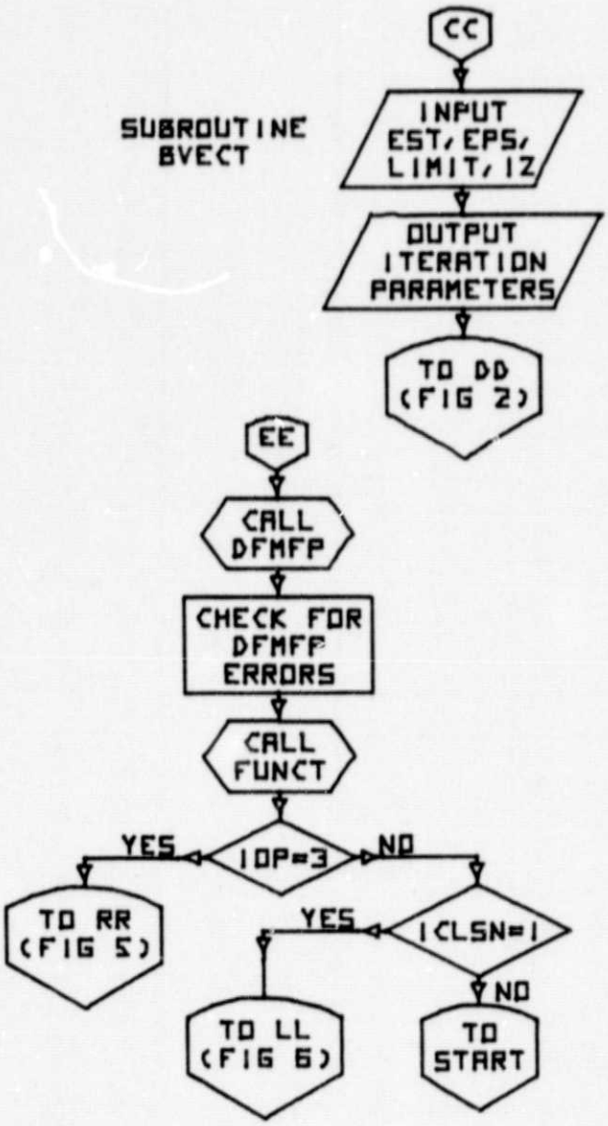


Fig. 3. Minimization of g(B) Flowchart.

3.4 Computation of $g(B)$ and $\frac{\partial g}{\partial B}$

The computation of $g(B)$ and $\frac{\partial g}{\partial 3}$ using the expressions given in Section 2.0 is performed in SUBROUTINE FUNCT (Fig. 4). The function subprogram DPHI computes

$$\phi(a) = \frac{1}{2} + \frac{1}{2} \text{ ERF } (a/\sqrt{2})$$

used in the computation of $g(B)$, where ERF is a library function subprogram given by

$$\text{ERF}(a) = 2(2\pi)^{-\frac{1}{2}} \int_0^a \exp\left[-\frac{1}{2} t^2\right] dt .$$

The transformed density functions $p_i(y,B)$, $1 \leq i \leq m$ are evaluated in the function subprogram XNDF.

In order to evaluate $g(B)$ and $\frac{\partial g}{\partial B}$, it is necessary to determine the regions $R_i(B)$, $1 \leq i \leq m$, defined in Section 2.0. An equivalent definition of the regions $R_i(B)$ is given by

$$R_i(B) = \{y \in E^1 : f_{ij}(y,B) \geq 0 \text{ for each } j\} , \quad 1 \leq i \leq m ,$$

where

$$f_{ij}(y,B) = \ln \frac{\alpha_i p_i(y,B)}{\alpha_j p_j(y,B)} , \quad i \neq j, \; i, j = 1, \ldots, m$$

$$= \eta_{ij}(B) y^2 + 2\beta_{ij}(B) y + \gamma_{ij}(B)$$

with

$$\eta_{ij}(B) = B\,\Sigma_i\,B^T - B\,\Sigma_j\,B^T$$

$$\beta_{ij}(B) = (B\,\Sigma_j\,B^T)B\mu_i - (B\,\Sigma_i\,B^T)B\mu_j$$

$$\gamma_{ij}(B) = (B\,\Sigma_i\,B^T)(B\mu_j)^2 - (B\,\Sigma_j\,B^T)(B\mu_i)^2$$

$$+ (B\,\Sigma_i\,B^T)(B\,\Sigma_j\,B^T)\,\ln\left(\frac{\alpha_i^2\,B\,\Sigma_j\,B^T}{\alpha_j^2\,B\,\Sigma_i\,B^T}\right).$$

To determine the regions $R_i(B)$, $1 \le i \le m$, the roots of the quadratic equations $f_{ij}(y,B) = 0$ are first determined. For the case where $\eta_{ij}(B) = 0$, a single root

$$y_{ij}(B) = \frac{(B\mu_j)^2 - (B\mu_i)^2 + \left(2\,\ln\frac{\alpha_i}{\alpha_j}\right)(B\,\Sigma_i\,B^T)}{2(B\mu_j - B\mu_i)}$$

is obtained which reduces to

$$y_{ij}(B) = \frac{B\mu_i + B\mu_j}{2}$$

when $\alpha_i = \alpha_j$. In the case where $\eta_{ij}(B) \neq 0$, the roots of $f_{ij}(y,B) = 0$ are computed using the quadratic formula.

Once all the roots $y_{ij}(B)$ of the quadratic equations $f_{ij}(y,B) = 0$, $i \neq j$, $i, j = 1, 2, \ldots, m$, have been found, we determine the regions $R_j(B)$, $1 \le j \le m$, by the following ordering defined on a (possibly proper) subset of the roots:

a) Choose i_1 such that

$$B \Sigma_{i_1} B^T = \max_{1 \le j \le m} B \Sigma_j B^T .$$

If

$$B \Sigma_j B^T = B \Sigma_k B^T = \max_{1 \le i \le m} B \Sigma_i B^T$$

for $j \neq k$, then choose i_1 such that

$$B\mu_{i_1} = \min_{j,k} \{B\mu_j, B\mu_k\}$$

b) Choose i_2 such that

$$y_{i_2 i_1} = \min_{1 \le j \le m} y_{j i_1}$$

c) Given i_1 and i_2, choose i_3 such that

$$y_{i_3 i_2} = \min_{1 \le j \le m} \{y_{j i_2} | y_{j i_2} > y_{i_2 i_1}\}$$

d) In general choose i_{k+1} such that

$$y_{i_{k+1} i_k} = \min_{1 \le j \le m} \{y_{j i_k} | y_{j i_k} > y_{i_k i_{k-1}}\} ,$$

provided

$$\{y_{j i_k} | y_{j i_k} > y_{i_k i_{k-1}}\} \neq \emptyset$$

If the above set is the null set, the procedure is terminated.

If in the above procedure,

$$y_{i_{k+1}i_k} = \min_{1\le j\le m} \{y_{ji_k} | y_{ji_k} > y_{i_k i_{k-1}}\}$$

for r choices $\{j_1,\ldots,j_r\}$ of i_{k+1}, then choose i_{k+1} such that

$$B\mu_{i_{k+1}} = \min_{1\le r\le s} \{B\mu_{j_s} | B\mu_{j_s} \ge y_{i_{k+1}i_k}\}$$

if the above set is nonempty; otherwise choose i_{k+1} such that

$$B\mu_{i_{k+1}} = \min_{1\le r\le s} B\mu_{j_s} .$$

The regions $R_j(B)$ are then given by

$$R_j(B) = \bigcup_{I_j} \{y | y_{i_k i_{k-1}} < y \le y_{i_{k+1}i_k}\} ,$$

where

$$I_j = \{i_k | i_k = j\} , \; j = 1, 2,\ldots,m .$$

For both the initial B vector and the final normalized B vector, SUBROUTINE FUNCT outputs $R_i(B)$, $B\,\Sigma_i\,B^T$, $B\mu_i$, $g(B)$, $\frac{\partial g}{\partial B}$, and B. For the final normalized B, the entries q_{ij} of the confusion matrix are computed from the formula

$$q_{ij} = \int_{R_i(B)} p_j(y,B)dy$$

and output.

The parameter IOUT is an internal output control flag provided to SUBROUTINE FUNCT.

IOUT . . . . . : Control flag

= -1 first pass (B_0)

= 0 intermediate iteration of DFMFP.

= 1 last pass (final B).

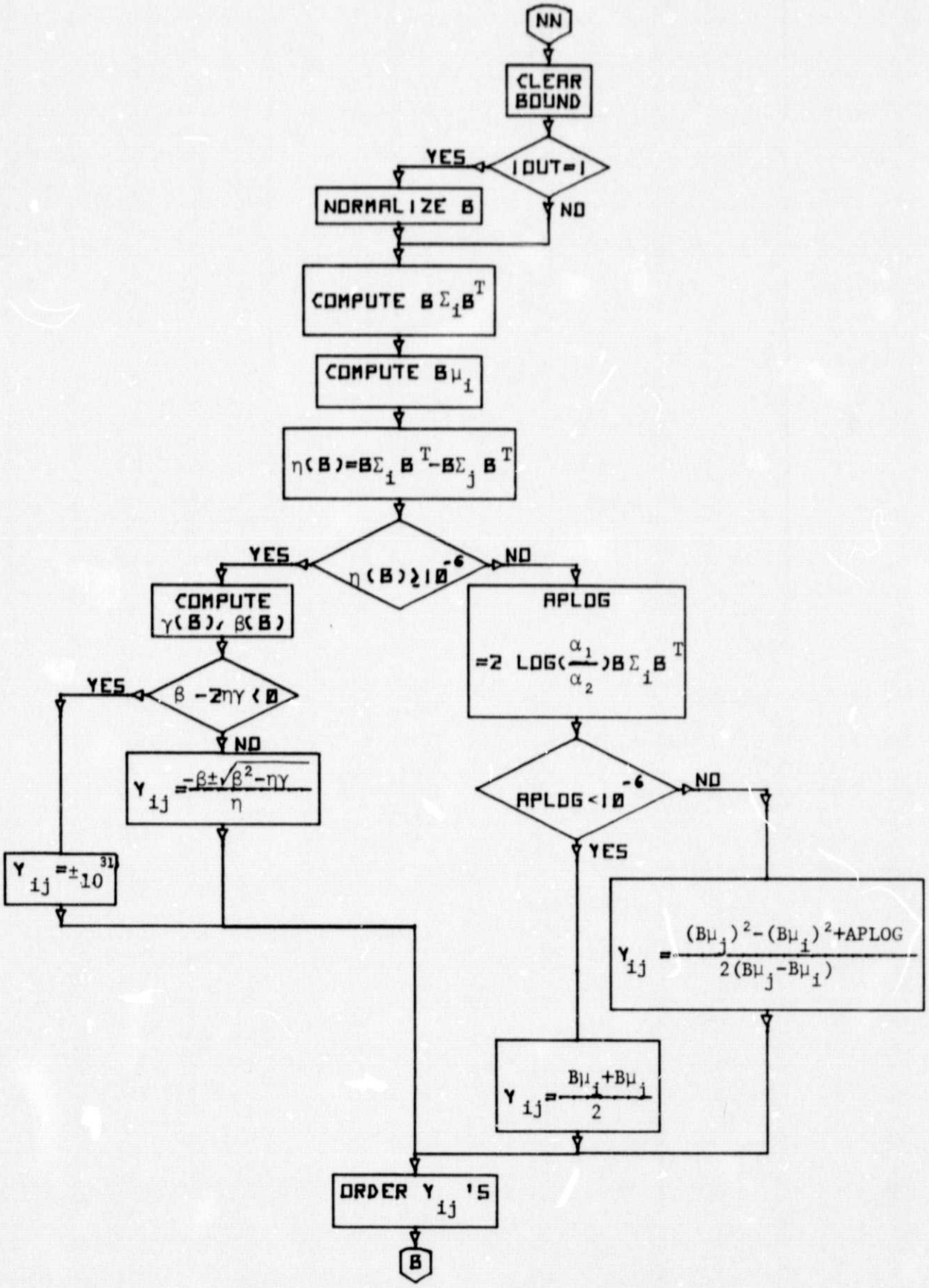


Fig. 4a. FUNCT Flowchart

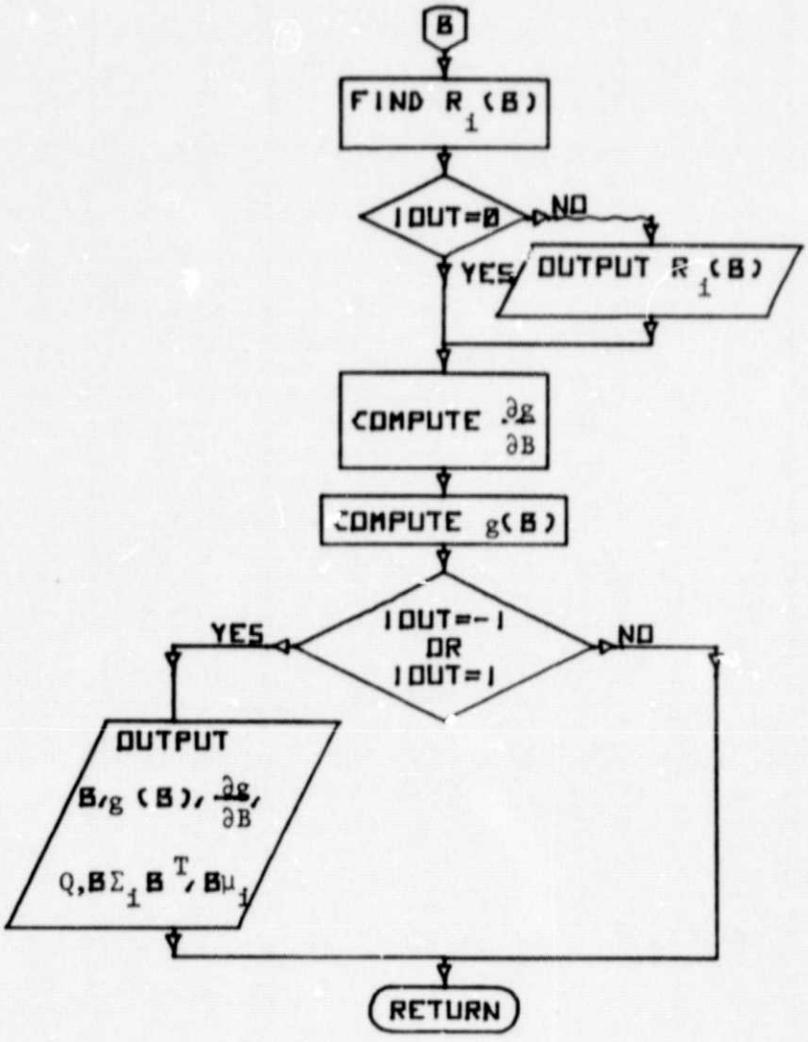


Fig. 4b. FUNCT Flowchart

3.5 Classification procedure

Provision is made for classification of labelled measurement vectors using the B vector and associated decision regions determined by the minimization procedure (Fig. 5). The format of the input data is provided by the user at object time. Input parameters are as follows:

NFPC . . . . . : number of features per class in the input vectors.

IO . . . . . : output flag.

= 0 no output of vectors.

= 1 output measurement vectors.

MI . . . . . : number of vectors per class for input data, single subscripted array.

IFEA . . . . . : numeric labels of the N desired features from the NFPC features in the measurement vectors, single subscripted array.

FMT . . . . . : format of the input data, single subscripted array.

If NFPC differs from N, the desired feature numbers are input. Each input vector is transformed to one dimension and classified according to the rule given in Section 2.0. An error matrix is constructed which counts the number of labelled vectors classified into each class. Output of the original classified vectors is under the control of the flag IO.

SUBROUTINE ERRMAT outputs an estimated confusion matrix computed from the error matrix generated by the classification routine. The ij^{th} entry is the fraction of the vectors from class j which are classified into class i. The percent of misclassification, defined as the percent of the total vectors which were misclassified, is computed and output. Control is then returned to the start of the parameter routine or else is given to SUBROUTINE RANCLN according to the value of ICLSN.

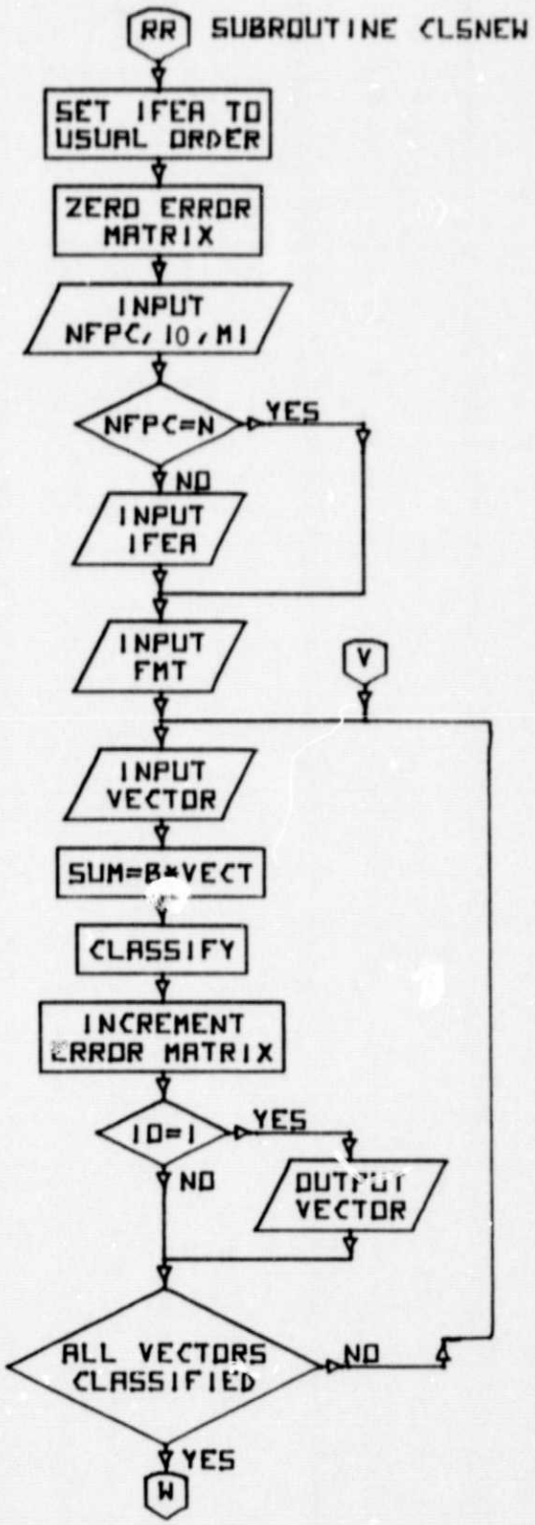


Fig. 5a. Classification Flowchart

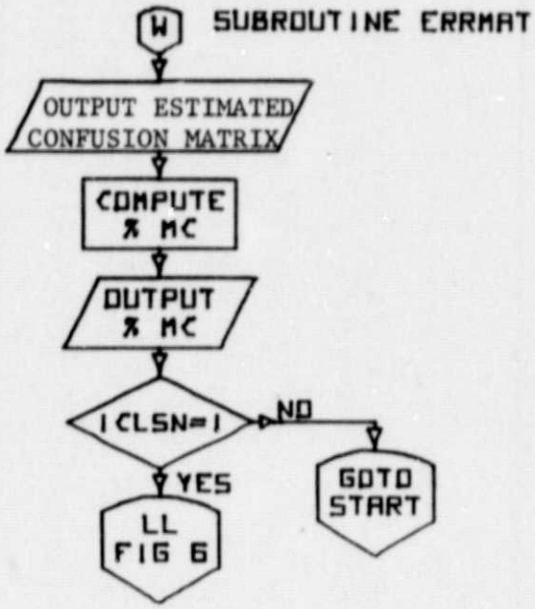


Fig. 5b. Classification Flowchart

3.6 Estimation of PMC

SUBROUTINE RANCLN (Fig. 6) provides an estimate of the probability of misclassification in the original feature space. Random vectors are generated using the class statistics of the original problem. The internal parameter NPC determines the number of vectors to be generated for each class. The user inputs the seed value for the random number generator, ISE, in order to insure the desired degree of randomness. SUBROUTINE GAUSS [10] computes a vector of N(0,I)-distributed random numbers using SUBROUTINE RANDU [10] to first generate a vector of uniformly distributed random numbers. Each vector so generated is used in the evaluation of the log of the density function for each class. SUBROUTINE MCHLSK (see [11]) computes the modified Cholesky decomposition of the covariance matrix in order to simplify the evaluation of the N-dimensional density function. The class whose density function times its a priori probability assumes the maximum value at the generated vector is assigned to that vector and the error matrix is incremented accordingly. The generated vector is printed if the value of the user entered parameter IO = 1. SUBROUTINE ERRMAT is called to print the estimated confusion matrix and to compute an estimate of the probability of misclassification and percent of misclassification.

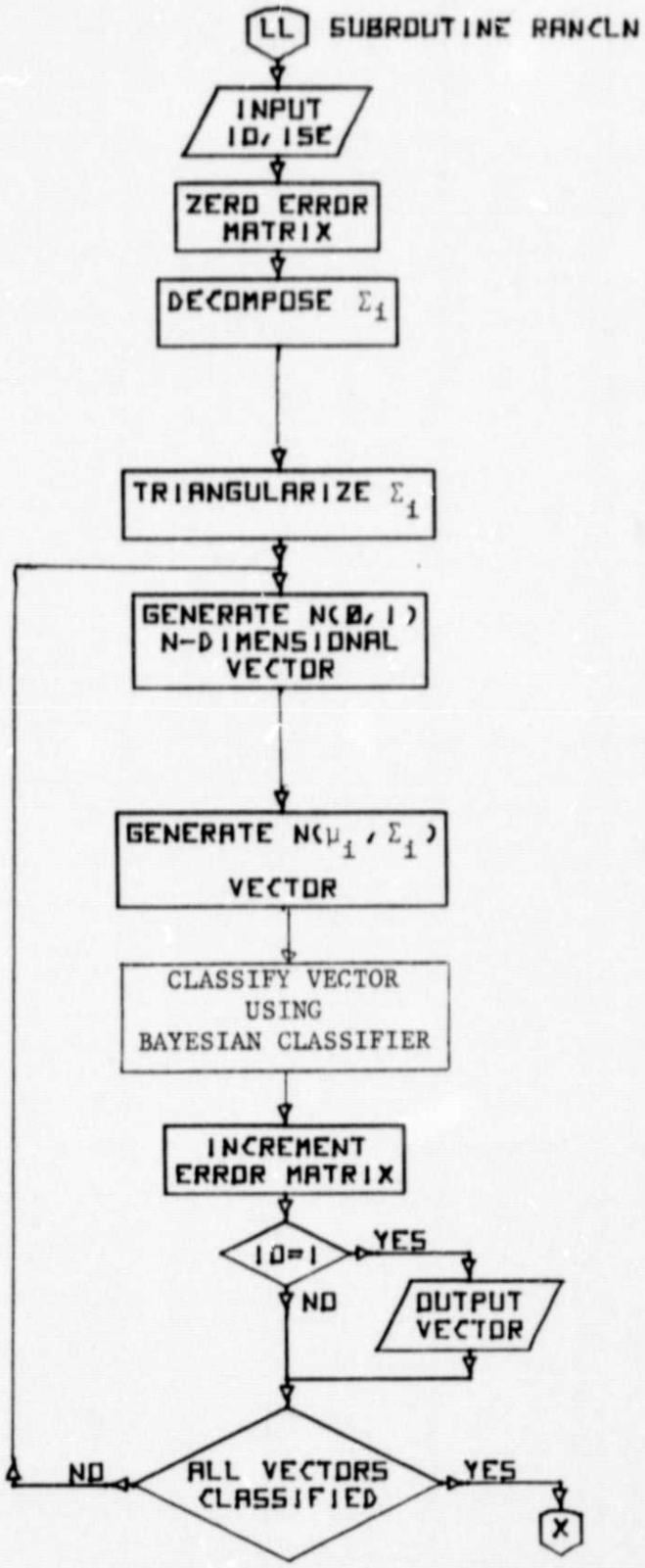


Fig. 6a. Estimation of PMC Flowchart

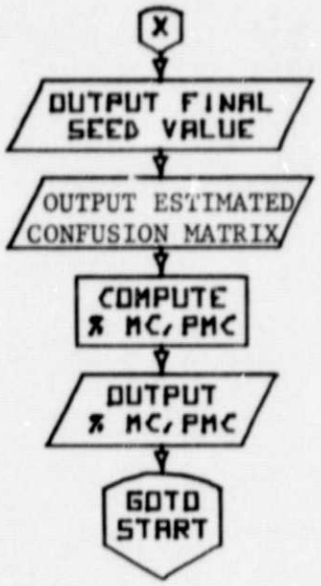


Fig. 6b. Estimation of PMC Flowchart

4.0 OPERATING PROCEDURE

In order to simulate object time dimensioning, the user must provide a calling routine of the following form:

```
DIMENSION ALRGE(IDIM)
DOUBLE PRECISION BLRGE(IDIM2)
COMMON MX,NX
MX =
NX =
CALL PRDIM(ALRGE,BLRGE)
STOP
END
```

The values of MX, NX, IDIM and IDIM2 are determined as follows:

MX = maximum value of MTOT for the program run.

NX = maximum value of NFPC for the program run.

$$IDIM = MX(4+MX+NX(\frac{NX+1}{2})) + NX(\frac{5}{2}NX + \frac{5}{2})$$

$$IDIM2 = MX(3+4MX+3NX+2NX^2) + NX(NX/2 + 11/2)$$

If available storage is not a problem, the user can incorporate maximum fixed dimensions into the program.

The program is suitable for interactive operation with the inclusion of parameter request messages. The program was written in IBM Fortran G with development on the Texas A&M University IBM 360-65.

Input parameters are of fixed format and must be in a specified order. The form and order are illustrated in Fig. 7. Sample input decks are presented in Section 5.0. In order to demonstrate the classification option, the standard iris data set from [2] is used.

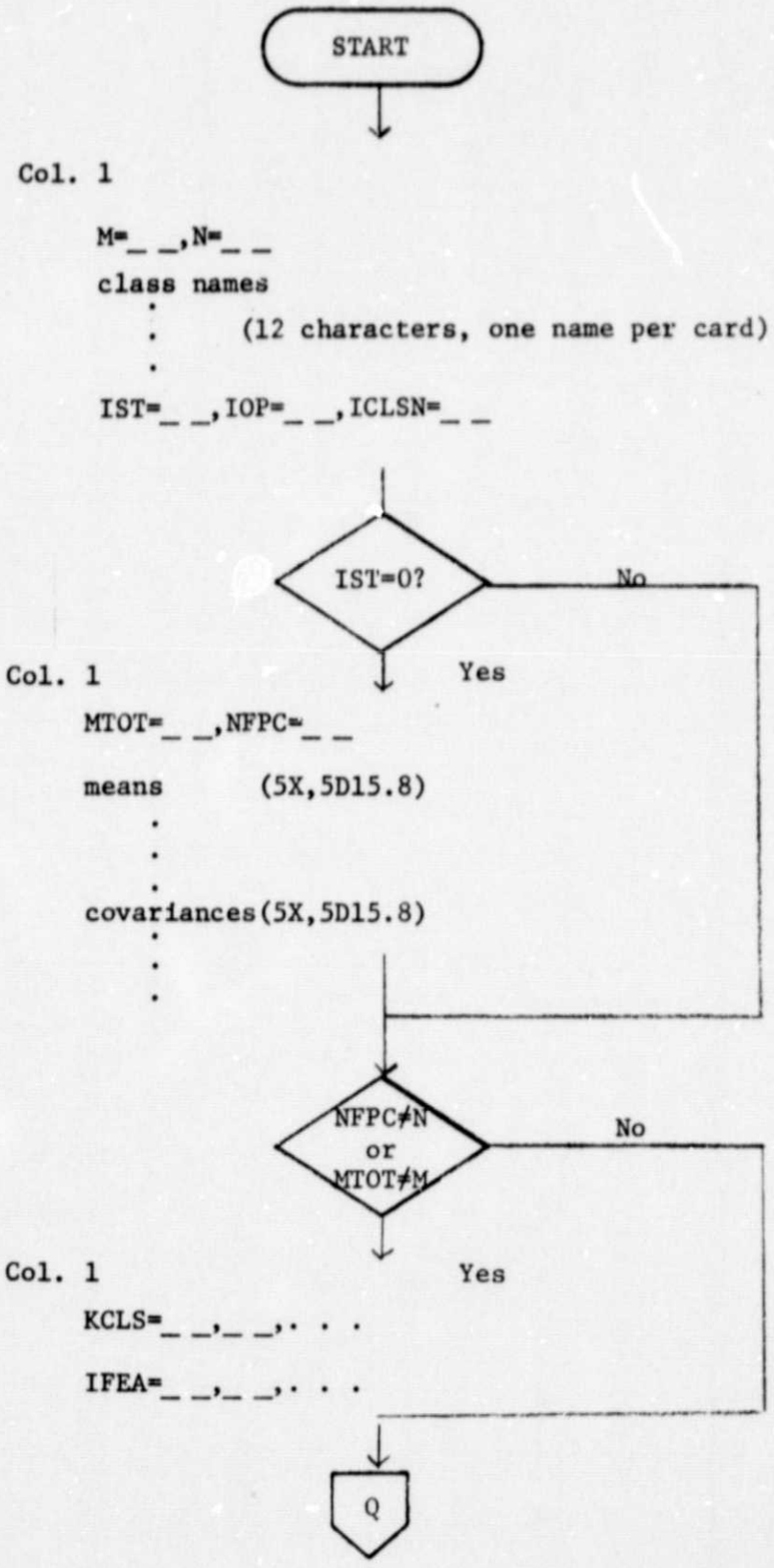


Fig. 7a. Input Parameter Deck Set-up

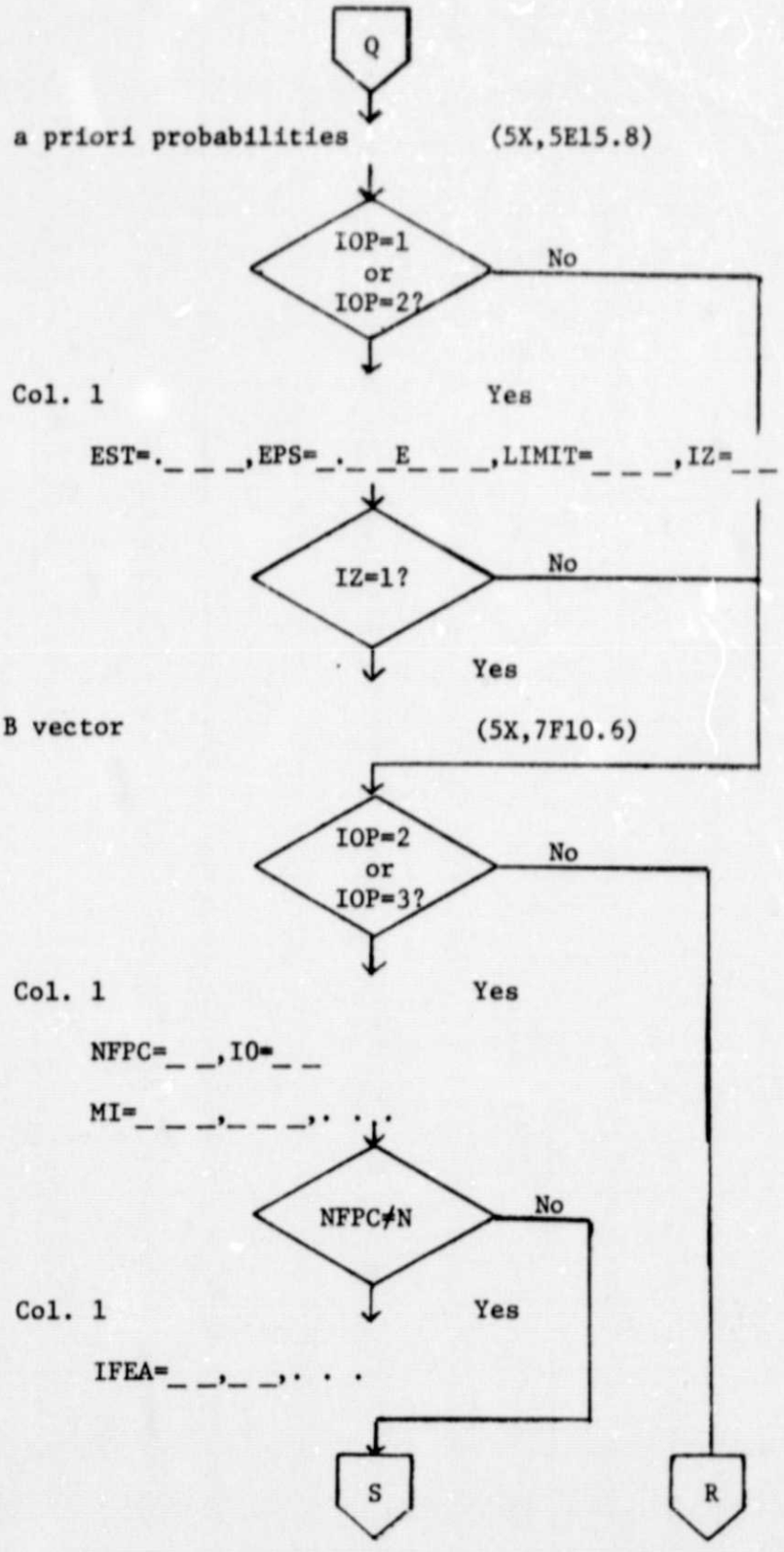


Fig. 7B. Input Parameter Deck Set-up

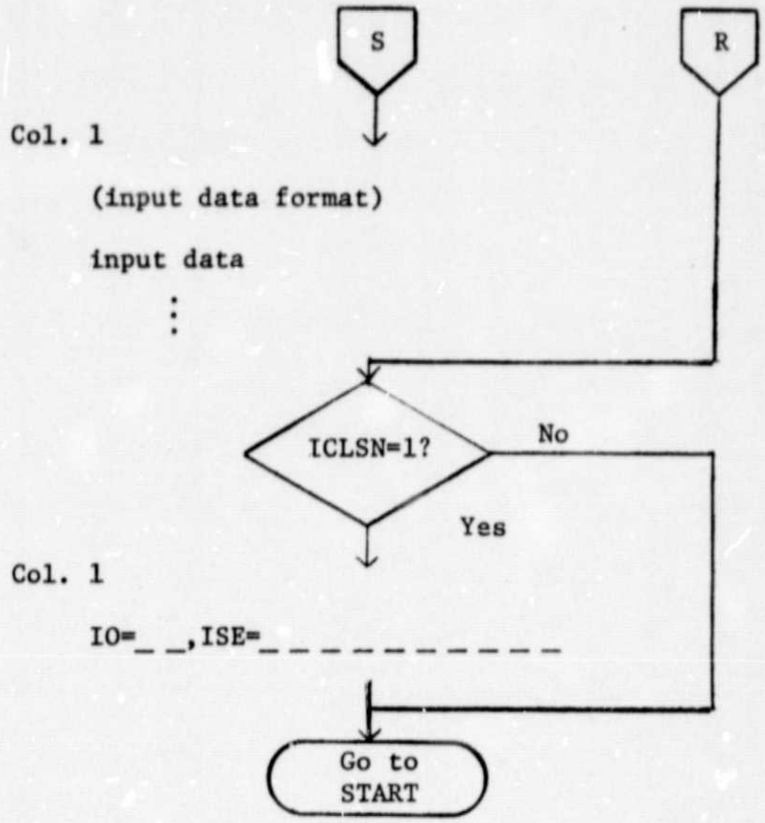


Fig. 7c. Input Parameter Deck Set-up

5.0 EXAMPLE INPUT DECKS

Example 1.

9 class Flight Line 210 data.

4 feature subset (9,11,2,10) of 12-dimensional statistics deck read in.

compute minimizing B vector.

compute estimated PMC in 4-dimensional feature space.

equal a priori probabilities.

compute B_0.

```
M=09,N=04
CLASS 1
CLASS 2
CLASS 3
CLASS 4
CLASS 5
CLASS 6
CLASS 7
CLASS 8
CLASS 9
IST=00,IOP=01,ICLSN=01
MTOT=09,NFPC=12
     statistics deck
KCLS=01,02,03,04,05,06,07,08,09
IFEA=09,11,02,10
     .11111111      .1111111       .1111111       .1111111       .1111111
     .11111111      .1111111       .11111111      .1111111
EST=.250,EPS=1.00E-04,LIMIT=050,IZ=00
IO=00,ISE=1110001110
```

Example 2.

9 class, 12-dimensional.

use previous statistics.

compute minimizing B vector.

compute estimated PMC in 12-dimensional feature space.

equal a priori probabilities.

compute B_0.

```
M=09,N=12
CLASS 1
CLASS 2
CLASS 3
CLASS 4
CLASS 5
CLASS 6
CLASS 7
CLASS 8
CLASS 9
IST=01,IOP=01,ICLSN=01
     .11111111      .11111111      .11111111      .11111111      .11111111
     .11111111      .11111111      .11111111      .11111111
EST=.250,EPS=1.00E-04,LIMIT=050,IZ=00
IC=00,ISE=0246897531
```

Example 3.

3-class, 12-dimensional.

use previous statistics.

compute minimizing B vector.

compute estimated PMC.

equal a priori probabilities.

compute B_0.

```
M=03,N=12
CLASS 1
CLASS 2
CLASS 3
IST=01,IOP=01,ICLSN=01
KCLS=01,02,03
IFEA=01,02,03,04,05,06,07,08,09,10,11,12
     .33333333      .33333333      .33333333
EST=.250,EPS=1.00E-04,LIMIT=050,IZ=00
IO=00,ISE=4562301852
```

Example 4.

9 class, 12 dimensional.

use previous statistics.

compute minimizing B vector.

compute estimated PMC.

unequal a priori probabilities.

compute B_0.

```
M=09,N=12
CLASS 1
CLASS 2
CLASS 3
CLASS 4
CLASS 5
CLASS 6
CLASS 7
CLASS 8
CLASS 9
IST=01,IOP=01,ICLSN=01
     .05            .05            .20            .10            .15
     .02            .08            .15            .20
EST=.250,EPS=1.00E-04,LIMIT=050,IZ=00
IO=00,ISE=1234567890
```

Example 5.

2 class, 12-dimensional.

use previous statistics

compute minimizing B vector.

equal a priori probabilities.

compute B_0.

```
M=02,N=12
CLASS 3
CLASS 4
IST=01,IOP=01,ICLSN=00
KCLS=03,04
IFEA=01,02,03,04,05,06,07,08,09,10,11,12
     .5             .5
EST=.250,EPS=1.00E-04,LIMIT=050,IZ=00
```

Example 6.

3 class iris data.

4-dimensional statistics deck read in.

compute minimizing B vector.

compute estimated PMC.

equal a priori probabilities.

read in B_0.

classify 10 measurement vectors from each class.

```
M=03,N=04
VERSICOLOR
SETOSA
VIRGINICA
IST=00,IOP=02,ICLSN=01
MTOT=03,NFPC=04
     statistics deck
     .33333333      .33333333      .33333333
EST=.250,EPS=1.00E-04,LIMIT=050,IZ=01
BVEC=0.0       1.0       0.0       1.0
NFPC=04,IO=01
MI=010,010,010
(4F2.1)
     measurement vectors
IO=00,ISE=9876543210
```

Example 7.

2 class, 4-dimensional.

use previous statistics

compute minimizing B vector.

compute estimated PMC.

equal a priori probabilities.

compute B_0.

```
M=02,N=04
VERSICOLOR
SETOSA
IST=01,IOP=01,ICLSN=01
KCLS=01,02
IFEA=01,02,03,04
     .5                .5
EST=.250,EPS=1.00E-04,LIMIT=050,IZ=00
IO=00,ISE=2369805804
```

6.0 SAMPLE OUTPUT

The sample output appearing on the following four pages was generated as a result of using the input deck given in Example 6 of Section 5.0.

COVARIANCE MATRIX FOR CLASS 1 - VERSICOLOR

0.266	0.085	0.183	0.056
0.085	0.098	0.083	0.041
0.183	0.083	0.221	0.073
0.056	0.041	0.073	0.039

MEAN VECTOR FOR CLASS 1 - VERSICOLOR

5.936	2.770	4.260	1.326

APRIORI PROBABILITY FOR CLASS 1= 0.333

COVARIANCE MATRIX FOR CLASS 2 - SETOSA

0.124	0.099	0.016	0.010
0.099	0.144	0.012	0.009
0.016	0.012	0.030	0.006
0.010	0.009	0.006	0.011

MEAN VECTOR FOR CLASS 2 - SETOSA

5.006	3.428	1.462	0.246

APRIORI PROBABILITY FOR CLASS 2= 0.333

COVARIANCE MATRIX FOR CLASS 3 - VIRGINICA

0.404	0.094	0.303	0.049
0.094	0.104	0.071	0.048
0.303	0.071	0.305	0.049
0.049	0.048	0.049	0.075

MEAN VECTOR FOR CLASS 3 - VIRGINICA

6.588	2.974	5.552	2.026

APRIORI PROBABILITY FOR CLASS 3= 0.333

ITERATION ROUTINE PARAMETERS

ESTIMATED MINIMUM= 0.25
ABSOLUTE ERROR= 1.0E-04
MAXIMUM NUMBER OF ITERATIONS= 50

RESULTS FOR INITIAL B VECTOR

REGIONS
-0.10000000000000000D 31 <Y< -0.36307274541736170 01 R 3(B)
-0.36307274541736170 01 <Y< 0.28037931408301210 00 R 1(B)
 0.28037931408301210 00 <Y< 0.39266945498254360 01 R 2(B)
 0.39266945498254360 01 <Y< 0.45530025522121350 01 R 1(B)
 0.45530025522121350 01 <Y< 0.10000000000000000D 31 R 3(B)

CLASS	TRANSFORMED COVARIANCES	TRANSFORMED MEANS
1	0.21998247131705280 00	0.40959890000000000D 01
2	0.17339095100760460 00	0.36739946000000000D 01
3	0.27469274401664730 00	0.49999870000000000D 01

PROB. OF MISCLASS. FOR CURRENT B= 0.33094727021160950 00

VECTOR OF GRADIENTS
FNTX(1) =-0.19816196239726590 00 FNTX(2) = 0.32129407162416530 00 FNTX(3) =-0.84085918670411020 00
FNTX(4) =-0.32129407162416850 00

B VECTOR
B(1) = 0.0 B(2) = 0.10000000000000000D 01 B(3) = 0.0
B(4) = 0.10000000000000000D 01

RESULTS FOR FINAL B VECTOR

REGIONS
-0.1[illegible] 31 <Y< -0.14709532954076010 02 R 3(B)
-0.14709532954076010 02 <Y< -0.50250766503778610 00 R 2(B)
-0.50250766503778610 00 <Y< 0.10418958173204900 01 R 1(B)
0.1 4189581732049D0 01 <Y< 0.10000000000000000 31 R 3(B)

CONFUSION MATRIX

0.972 0.000 0.031

0.000 1.000 0.000

0.028 0.0 0.969

CLASS	TRANSFORMED COVARIANCES	TRANSFORMED MEANS
1	0.51908399297150220-01	0.60631175592269090 00
2	0.41223844840150190-01	-0.14954337654656640 01
3	0.62087254362810910-01	0.15064599936808370 01

PROB. OF MISCLASS. FOR CURRENT B= 0.19693581547611980-01

VECTOR OF GRADIENTS
FNTX(1) = 0.84795814403542700-07 FNTX(2) = 0.33159180027603490-07 FNTX(3) =-0.45051681857311010-07
FNTX(4) = 0.62275487808238770-07

B VECTOR
B(1) =-0.21716654989843030 00 B(2) =-0.35997568164437960 00 B(3) = 0.43036921688463900 00
B(4) = 0.79877314388252900 00

CLASS - 1	CLASSIFIED AS - 1		
7.000	3.200	4.700	1.400
CLASS - 1	CLASSIFIED AS - 1		
6.400	3.200	4.500	1.500
CLASS - 1	CLASSIFIED AS - 1		
6.900	3.100	4.900	1.500
CLASS - 1	CLASSIFIED AS - 1		
5.500	2.300	4.000	1.300
CLASS - 1	CLASSIFIED AS - 1		
6.500	2.800	4.600	1.500
CLASS - 1	CLASSIFIED AS - 1		
5.700	2.800	4.500	1.300
CLASS - 1	CLASSIFIED AS - 1		
6.300	3.300	4.700	1.600
CLASS - 1	CLASSIFIED AS - 1		
4.900	2.400	3.300	1.000
CLASS - 1	CLASSIFIED AS - 1		
6.600	2.900	4.600	1.300
CLASS - 1	CLASSIFIED AS - 1		
5.200	2.700	3.900	1.400
CLASS - 2	CLASSIFIED AS - 2		
5.000	3.000	1.600	0.200
CLASS - 2	CLASSIFIED AS - 2		
5.000	3.400	1.600	0.400
CLASS - 2	CLASSIFIED AS - 2		
5.200	3.500	1.500	0.200
CLASS - 2	CLASSIFIED AS - 2		
5.200	3.400	1.400	0.200
CLASS - 2	CLASSIFIED AS - 2		
4.700	3.200	1.500	0.200

CLASS - 2 CLASSIFIED AS - 2
4.800 3.100 1.600 0.200

CLASS - 2 CLASSIFIED AS - 2
5.400 3.400 1.500 0.400

CLASS - 2 CLASSIFIED AS - 2
5.200 4.100 1.500 0.100

CLASS - 2 CLASSIFIED AS - 2
5.500 4.200 1.400 0.200

CLASS - 2 CLASSIFIED AS - 2
4.900 3.100 1.500 0.200

CLASS - 3 CLASSIFIED AS - 3
6.300 3.300 6.000 2.500

CLASS - 3 CLASSIFIED AS - 3
7.200 3.200 6.000 1.800

CLASS - 3 CLASSIFIED AS - 3
6.200 2.800 4.800 1.800

CLASS - 3 CLASSIFIED AS - 3
6.100 3.000 4.900 1.800

CLASS - 3 CLASSIFIED AS - 3
6.400 2.800 5.600 2.100

CLASS - 3 CLASSIFIED AS - 3
7.200 3.000 5.800 1.600

CLASS - 3 CLASSIFIED AS - 3
7.400 2.800 6.100 1.900

CLASS - 3 CLASSIFIED AS - 3
7.900 3.800 6.400 2.000

CLASS - 3 CLASSIFIED AS - 3
6.400 2.800 5.600 2.200

CLASS - 3 CLASSIFIED AS - 1
6.300 2.800 5.100 1.500

ESTIMATED CONFUSION MATRIX
FOR TEST DATA

1.000 0.0 0.100

0.0 1.000 0.0

0.0 0.0 0.900

PERCENT OF MISCLASSIFICATION= 3.33%

FINAL SEED VALUE(RANDOM NUMBER GENERATOR) - 1361779434

ESTIMATED CONFUSION MATRIX
FOR RANDOMLY GENERATED VECTORS IN N DIMENSIONS

0.970 0.0 0.025

0.0 1.000 0.0

0.030 0.0 0.975

PERCENT OF MISCLASSIFICATION= 1.83%

ESTIMATED PMC=0.018

REFERENCES

1. T. W. Anderson, An Introduction to Multivariate Statistical Analysis, John Wiley and Sons, New York, 1958.
2. R. A. Fischer, The use of multiple measurements in taxonomic problems, Ann. Eugenics 7(1936), 179-188.
3. R. Fletcher and M. J. P. Powell, A rapidly convergent descent method for minimization, Comput. J. 6(1963), 163-168.
4. L. F. Guseman, Jr., and Homer F. Walker, On minimizing the probability of misclassification for linear feature selection, JSC Internal Technical Note, JSC-08412, Johnson Space Center, Houston, Texas, August, 1973.
5. L. F. Guseman, Jr., B. Charles Peters, Jr. and Homer F. Walker, On minimizing the probability of misclassification for linear feature selection, Ann. Statist. (to appear).
6. L. F. Guseman, Jr., and Bruce P. Marion, Obtaining initial vectors for minimizing the probability of misclassification, Contract Report #1, NAS-9-13894, Texas A&M University, Department of Mathematics, Sept. 1974.
7. D. G. Luenberger, Optimization by Vector Space Methods, John Wiley and Sons, New York, 1969.
8. W. Robert Mann, Mean value methods in iteration, Proc. Amer. Math. Soc. 4(1953), 506-510.
9. J. L. Solomon and B. Charles Peters, Jr., A simplified version of locating stationary points of the transformed probability of correct classification, Preprint.

10. System/360 Scientific Subroutine Package, Version III, 5th Edition, Form GH20-0205-4, International Business Machines, August, 1970.

11. D. L. Van Rooy, M. S. Lynn, and C. H. Snyder, The use of the modified Cholesky decomposition in divergence and classification calculations, ICSA-275-025-008, Institute for Computer Sciences and Applications, Rice University, May, 1973.